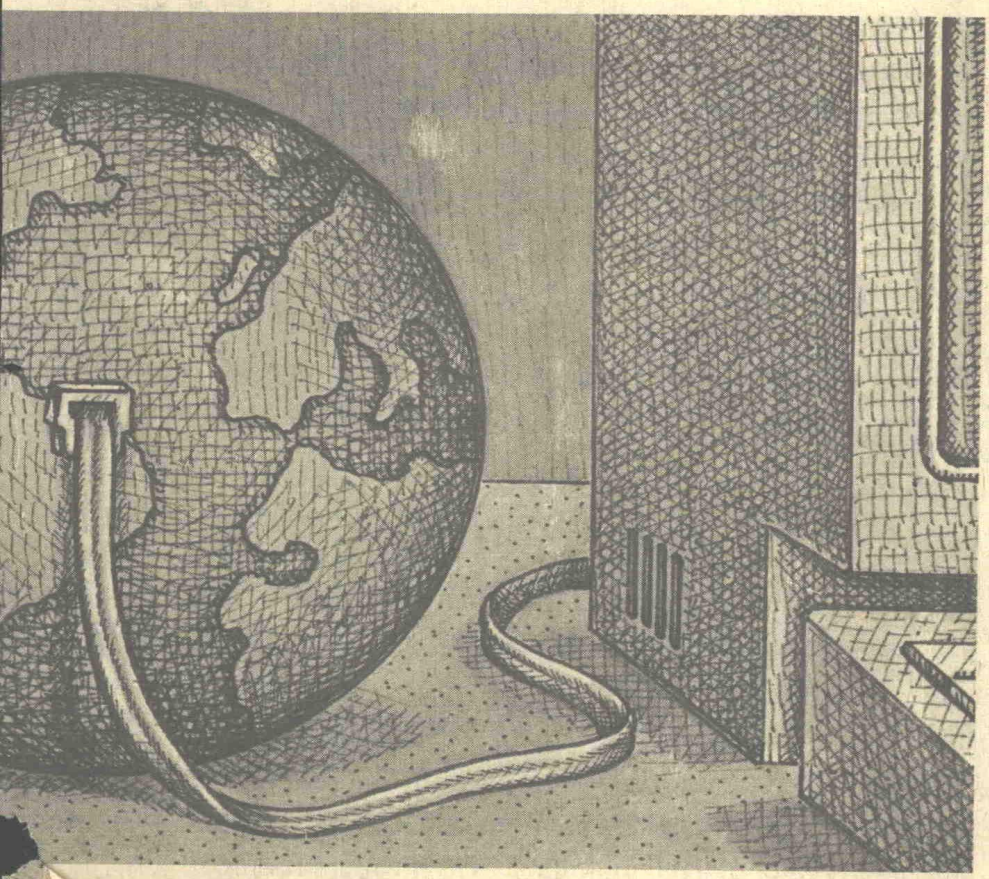
บทความพิเศษ

High Performance Computer

ยืน ภู่วรวรรณ

คอมพิวเตอร์ประสิทธิภาพสูง

หลักการของเทคโนโลยีที่จะทำให้คอมพิวเตอร์พัฒนาพ้นภาวะข้อจำกัดที่ทำงานอยู่ภายใต้ตัวชิพ



ในช่วงสิบปีที่ผ่านมา มีการกล่าวถึงคอมพิวเตอร์ประสิทธิภาพสูงกันมาก แต่เดิมเราแบ่งแยกคอมพิวเตอร์เป็นไมโครคอมพิวเตอร์ มินิคอมพิวเตอร์ เมนเฟรม และซูเปอร์คอมพิวเตอร์

คำว่าซูเปอร์คอมพิวเตอร์ดูจะไกลตัวผู้ใช้อยู่มาก แต่ในปัจจุบันสถาปัตยกรรมคอมพิวเตอร์ได้พัฒนาไปไกลมาก ประจวบกับการพัฒนาคอมพิวเตอร์ให้มีซีพียูจำนวนหลายตัวทำงานขนานกัน แนวโน้มการใช้คอมพิวเตอร์แบบขนานที่มีประสิทธิภาพสูงจึงเป็นหนทางหนึ่งที่ผู้ใช้จะเลือกใช้ได้ และการพัฒนาสถาปัตยกรรมแนวนี้จึงเป็นที่มาของคอมพิวเตอร์ประสิทธิภาพสูง หรือที่เรียกว่า HPC – High Performance Computer

สถาปัตยกรรมแบบขนาน

เพื่อให้คอมพิวเตอร์ทำงานที่ประสิทธิภาพสูง จำเป็นที่จะต้องสร้างกลไกการทำงานพิเศษ ทั้งนี้เพราะเทคโนโลยีหลายด้านมีขอบเขตจำกัด เช่น ถ้าใช้ความเร็วสัญญาณนาฬิกาก็มีความเร็วที่จำกัด เช่น ใช้ด้วยความเร็ว 66 เมกะเฮิรตซ์ หากเพิ่มความเร็วขึ้นอีกก็ทำได้ยากขึ้นและความเร็วก็คงไม่สูงมากนัก หนทางที่จะเพิ่มประสิทธิภาพจึงอยู่ที่สถาปัตยกรรมพิเศษเพื่อให้การทำงานได้ประสิทธิภาพสูงขึ้น การดำเนินการเช่นนี้จึงต้องใช้วิธีการที่เรียกว่า การทำงานแบบขนาน การพัฒนาสถาปัตยกรรมจึงสร้างบนแนวทางต่อไปนี้

การวางไปป์ไลน์คำสั่ง (Instruction pipelining)

เป็นวิธีการที่ซีพียูรุ่นใหม่หนีไม่พ้น ซีพียูที่ใช้โครงสร้าง RISC ทุกตัวใช้หลักการนี้ คือให้ชุดคำสั่งมีไม่มาก แต่มีโครงสร้างลำดับการทำงานคล้ายกัน เพื่อว่าจะได้วางลำดับคำสั่งแบบขนานเหลื่อมกันได้ การใช้ไปป์ไลน์สามารถทำงานได้มากคำสั่งในจำนวนสัญญาณนาฬิกาที่กำหนดให้ RISC ส่วนใหญ่ทำงานได้เร็วด้วยวิธีนี้ เพราะเป็นวิธีที่เพิ่มความเร็วได้ชัดเจน ซีพียูรุ่นใหม่ เช่น 80386, 80486 หรือแม้แต่เพนเทียมก็ใช้โครงสร้างการวางไปป์ไลน์เพื่อเพิ่มความเร็วในการทำงาน

การใช้ซีพียูหลาย ๆ ตัวทำงานขนานกัน

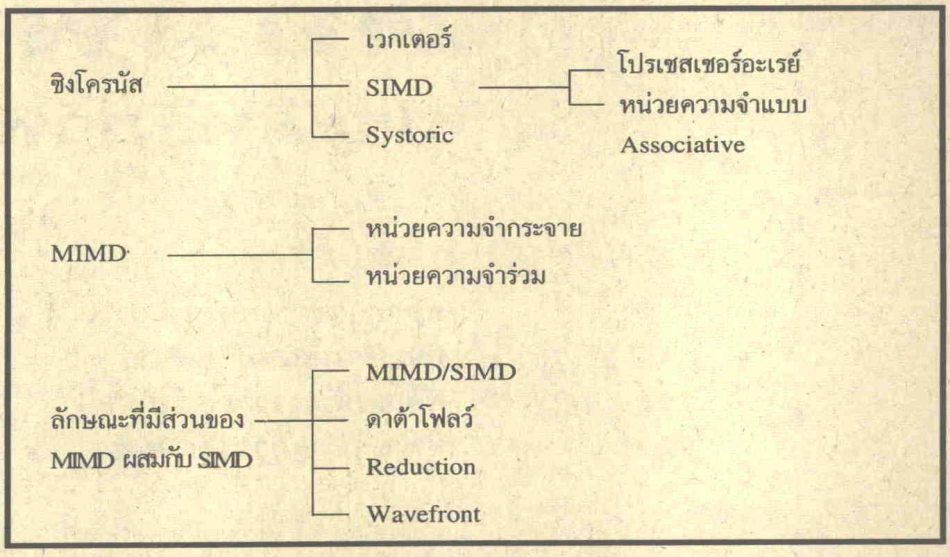
เป็นการใช้ซีพียูที่มีกลไกการทำงานอิสระ แต่ละตัวทำงานร่วมกัน การใช้ซีพียูจำนวนมากเป็นแนวความคิดที่ให้ซีพียูช่วยกันทำงาน เพื่อว่าในช่วงเวลาหนึ่งจึงทำงานได้หลายอย่างพร้อมกัน และจะเป็นผลทำให้ประสิทธิภาพการทำงานสูงขึ้น

การแยกส่วนการทำงานระหว่างซีพียูกับอินพุตเอาต์พุต

ก็เป็นอีกวิธีหนึ่งที่จะเพิ่มประสิทธิภาพการทำงาน โครงสร้างการจัดการอุปกรณ์เพอร์เฟอรัลมีลักษณะพิเศษ การแยกตัวประมวลผล I/O ออกไปทำให้ภาระของซีพียูหลักลดลงไปมาก ทำให้การทำงานโดยรวมดีขึ้น

โครงสร้างสถาปัตยกรรมของ HPC

HPC – High Performance Computer มีหลายสถาปัตยกรรม รูปที่ 1 แสดงไดอะแกรมที่เป็นไปได้ในเชิงสถาปัตยกรรมที่นำมาใช้กับ HPC HPC ส่วนใหญ่จะเข้าอยู่ในสถาปัตยกรรมใดสถาปัตยกรรมหนึ่ง หรือผสมกันระหว่างสถาปัตยกรรมเหล่านี้



รูปที่ 1 ไดอะแกรมสถาปัตยกรรมที่ใช้กับ HPC

HPC จึงเป็นคอมพิวเตอร์ที่เน้นการทำงานให้ได้ประสิทธิภาพสูง จึงมีโครงสร้างที่เหมาะสมกับการประยุกต์ใช้งานแตกต่างกัน งานบางอย่างใช้ได้ดีกับสถาปัตยกรรมหนึ่งแต่อาจไม่เหมาะกับอีกสถาปัตยกรรมหนึ่ง

ซิงโครนัสหมายถึงการใช้สัญญาณนาฬิกาเดียวกัน

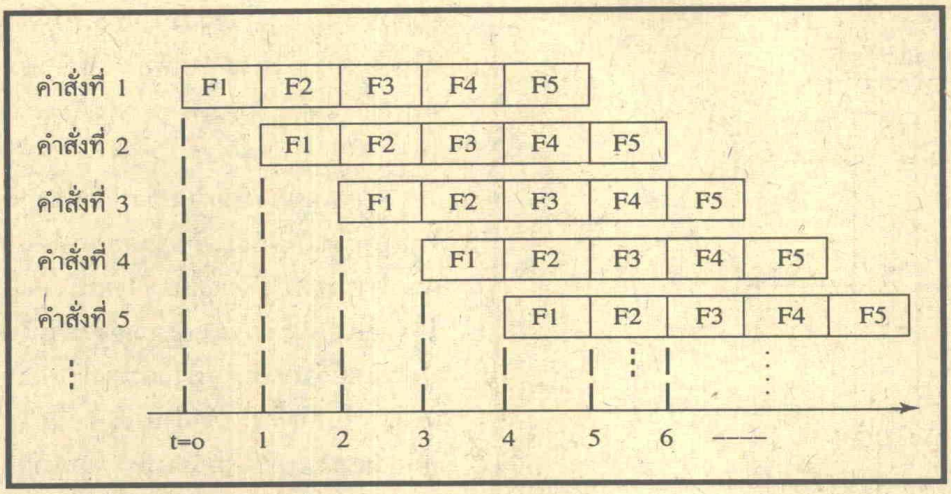
การทำงานที่ใช้สัญญาณนาฬิการ่วมกันหรือชุดเดียวกันเราเรียกว่าสถาปัตยกรรมซิงโครนัส การควบคุมอาจใช้ระบบการควบคุมร่วมกันจากส่วนกลางเพื่อควบคุมการทำงานแบบต่าง ๆ สถาปัตยกรรมแบบนี้เป็นแบบที่สร้างเพื่อให้มีส่วนร่วมกัน เพื่อลดความยุ่งยากซับซ้อนของการสร้างลง ในส่วนของซิงโครนัสแบ่งแยกออกเป็นสามกลุ่มคือ เวกเตอร์ SIMD และซิสโตริก

เวกเตอร์โปรเซสเซอร์

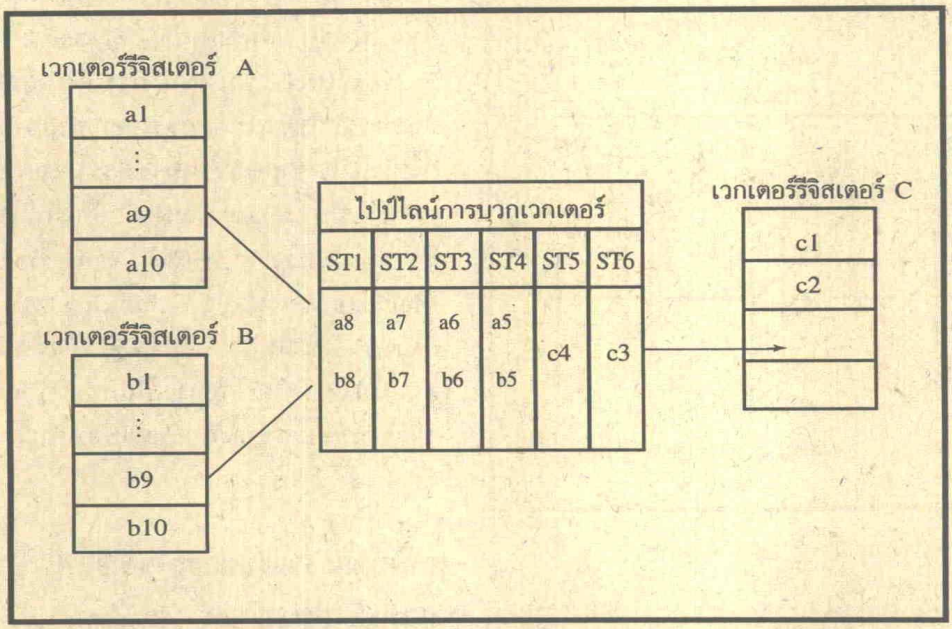
สถาปัตยกรรมที่ใช้หลักการเวกเตอร์โปรเซสเซอร์ได้รับการพัฒนาตั้งแต่ปี พ.ศ. 2515 ทั้งนี้ต้องการสร้างสถาปัตยกรรมที่เหมาะกับการประมวลผลเวกเตอร์เพื่อการคำนวณแมตริกซ์ การคำนวณมีลักษณะเป็นไปป์ไลน์เพื่อทำงานทางคณิตศาสตร์และลอจิก โครงสร้างไปป์ไลน์นี้ทำให้ทำงานพร้อม ๆ กันได้หลายงาน และสามารถทำงานได้ทั้งรูปแบบเป็นสเกล่าร์หรือเวกเตอร์

รูปที่ 2 เป็นโครงสร้างคำสั่งที่ใช้ไปป์ไลน์ โดยแบ่งคำสั่งออกเป็นส่วนการทำงานย่อย ๆ หลาย ๆ งาน วางโครงสร้างการทำงานต่อเนื่องกันไป โดยแต่ละคำสั่งที่ทำเรียงกันไปต้องมีรูปแบบการทำงานเหมือนกัน

สำหรับเวกเตอร์นั้นเน้นการทำงานที่เหมือนกันเป็นกลุ่ม เช่น การกระทำแมตริกซ์ที่ต้องมีการกระทำซ้ำ ๆ กันจำนวนมาก แต่ใช้ข้อมูลต่างกัน เช่น



รูปที่ 2 โครงสร้างคำสั่งไปป์ไลน์ที่ใช้ในซีพียูประเภท RISC



รูปที่ 3 การทำงานแบบเวกเตอร์โดยใช้รีจิสเตอร์ภายในซีพียู

รูปที่ 3 เป็นการวางไปป์ไลน์ 6 ขั้นของการบวกเวกเตอร์ โดยมี A1 – A10 บวกกับ B1 – B10 ได้ผลลัพธ์เป็น C1 – C10 กล่าวคือ A1 + B1 → C1, ...A10 + B10 → C10 การกระทำในลักษณะเวกเตอร์เป็นการโหลดตัวเลขทั้งหมดเข้าสู่รีจิสเตอร์ภายในซีพียู ดังนั้นซีพียูจะต้องมีรีจิสเตอร์จำนวนมากเพื่อรองรับการทำงานแบบเวกเตอร์

รูปที่ 3 เป็นตัวอย่างของการทำงานแบบเวกเตอร์โดยอาศัยรีจิสเตอร์ภายในของซีพียูเป็นบัฟเฟอร์เพื่อทำให้เกิดการทำงานแบบขนานร่วมกัน

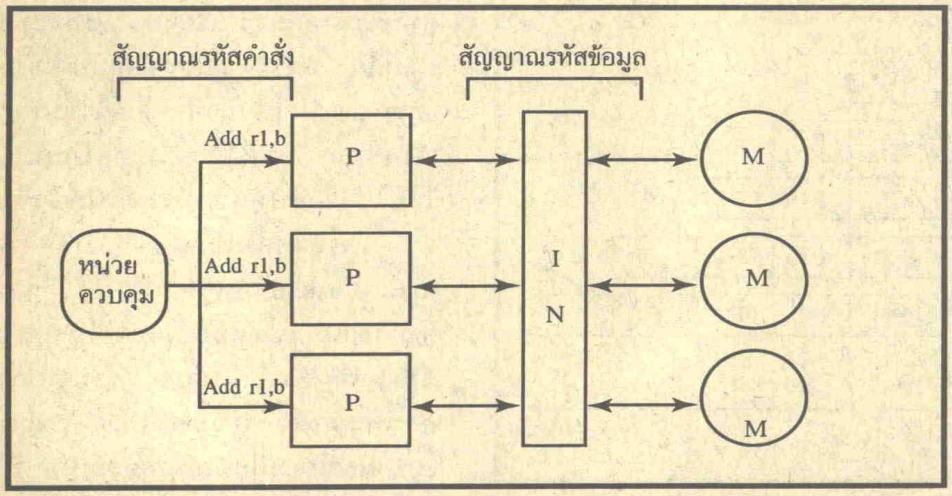
หากคำสั่งในการบวกตามรูปที่ 3 ต้องใช้เวลา 120 นาโนวินาที แต่การทำงานแบบเวกเตอร์มีไปป์ไลน์แบบ 6 ขั้นเพื่อทำการบวกเรียงตัวเลขทีละตัว ผลก็คือ แต่ละตัวเลขจะได้ผลลัพธ์ในเวลา 20 นาโนวินาที หรือเร็วขึ้นจากเดิมหกเท่า การบวกเวกเตอร์แบบใช้ไปป์ไลน์จึงทำให้การทำงานทำได้เร็วขึ้น

เครื่องคอมพิวเตอร์ Cray – 1 และ Fujitsu VP – 200 ใช้วิธีการคำนวณแบบเวกเตอร์ โดยใช้รีจิสเตอร์ภายในเป็นตัวดำเนินการ ส่วนซูเปอร์คอมพิวเตอร์ของบริษัท CDC Cyber 205 และของเท็กซัสอินสตรูเมนต์ ใช้เวกเตอร์อยู่ที่หน่วยความจำ

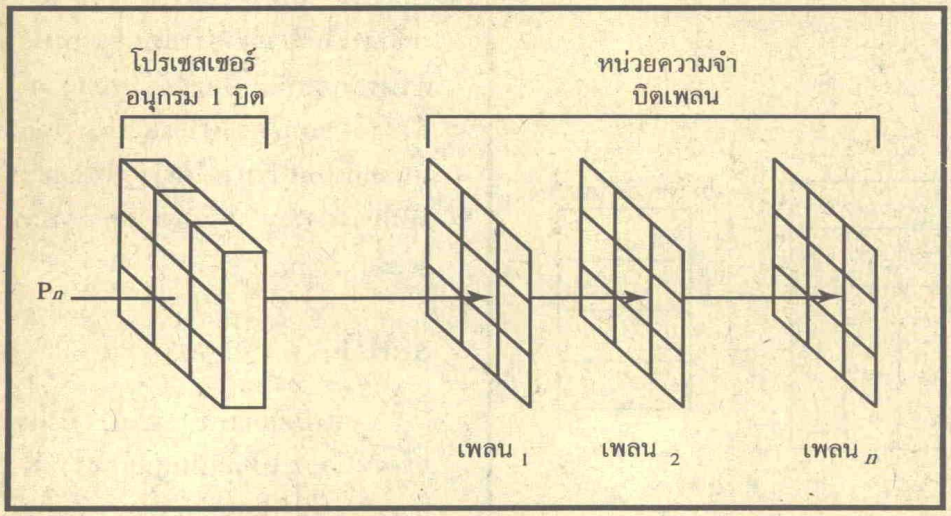
ซูเปอร์คอมพิวเตอร์รุ่นใหม่ ๆ เช่น Cray X – MP/4 และ ETA – 10 ใช้หลักการให้มีเวกเตอร์โปรเซสเซอร์หลายตัวประกอบกัน โดยใช้จำนวนจาก 4 ถึง 10 เวกเตอร์โปรเซสเซอร์ โดยใช้หน่วยความจำขนาดใหญ่ร่วมกันและยังให้มีการทำงานในลักษณะของ MIMD ได้

SIMD

สถาปัตยกรรม SIMD มีโครงสร้างหน่วยควบคุมเป็นศูนย์กลางหน่วยเดียว แต่มีหน่วยประมวลผลหลาย ๆ หน่วยร่วมกันทำงาน มีการเชื่อมต่อกับหน่วยความจำและเชื่อมต่อระหว่างโปรเซสเซอร์เป็นเครือข่าย การสื่อสารส่วนนี้สามารถส่งผ่านข้อมูลระหว่างหน่วยประมวลผลและระหว่างหน่วยประมวลกับหน่วยความจำ ศูนย์กลางการควบคุมอยู่ที่หน่วยควบคุมที่จะทำการกระจายคำสั่งไปยังโปรเซสเซอร์ต่าง ๆ เพื่อให้ทุกโปรเซสเซอร์ทำงานไปด้วยกัน การทำงานมีการส่งข้อมูลหรือสร้างลำดับการทำงานระหว่างโปรเซสเซอร์ เครือข่ายที่เชื่อมโยงระหว่างโปรเซสเซอร์จึงเป็นสิ่งสำคัญที่จะทำให้ประสิทธิภาพการ



รูปที่ 4 การทำงานแบบ SIMD



รูปที่ 5 การจัดวางเป็นบิตเพลนขนาด 64x64

ทำงานดียิ่งขึ้น การจัดการในส่วนเครือข่ายของโปรเซสเซอร์มีรูปแบบได้หลายรูปแบบ

สถาปัตยกรรมแบบโปรเซสเซอร์อะเรย์

เป็นสถาปัตยกรรมที่เหมาะสมกับการทำงานทางด้านวิทยาศาสตร์ เช่น การคำนวณรูปภาพ การประมวลผลภาพ การสร้างแบบจำลองทางคณิตศาสตร์ การคำนวณในส่วนที่มีตัวข้อมูลจำนวนมากและข้อมูลแบ่งเป็นชิ้นเล็ก ๆ ที่แยกการคำนวณได้ โปรเซสเซอร์อะเรย์ได้รับการพัฒนามาตั้งแต่ปี พ.ศ. 2503 เช่น เครื่อง Illiac – IV หรือเครื่องขนาดใหญ่ในยุคหลัง เช่น Burrought การคำนวณจึงต้องมีการเชื่อมโยงระหว่างหน่วยความจำกับโปรเซสเซอร์และโปรเซสเซอร์ด้วยกันเอง เป็นเสมือนตารางครอสกันไปมา ลักษณะเช่นนี้เป็นโมเดลการจัดวางโปรเซสเซอร์ที่เหมาะสม

โปรเซสเซอร์อะเรย์ที่นิยมอย่างหนึ่งคือ การนำเอาโปรเซสเซอร์ขนาด 1 บิตจำนวนมากมาจัดวางเป็นอะเรย์ โดยเป็นบิตเพลนในระดับเดียวกัน เช่น ขนาด 64×64 เป็นเพลน ดังตัวอย่างในรูปที่ 5

ขนาดของหน่วยความจำก็จะทำในลักษณะเป็นอะเรย์เช่นเดียวกันด้วย การทำงานของโปรเซสเซอร์จึงดำเนินการได้ทั้งในระดับอะเรย์ของตัวเอง หรือในหน่วยความจำที่อยู่ต่างระดับ การคำนวณในลักษณะนี้จึงให้ประสิทธิภาพที่ดีกว่าเพราะทำได้พร้อมกัน สามารถจัดวางโครงสร้างข้อมูลได้หลายรูปแบบ

หลายบริษัทได้ออกแบบสถาปัตยกรรมให้อยู่ในรูปของโปรเซสเซอร์ขนานจำนวนมาก ที่เรียกว่า Massively Parallel Processor เช่น บริษัท Loral และ ICL ใช้สถาปัตยกรรมการกระจายเพื่อให้โปรเซสเซอร์ทำการประมวลผลแต่ละพิกเซลของภาพในหน่วยความจำได้อย่างรวดเร็ว ส่วน CM-Connection Machine ของบริษัท Thinking Machines ใช้ซีพียูถึง 65536 ตัว แต่ละตัวมีขนาด 1 บิต โดยวางเป็นกลุ่ม ๆ ละ 4 ตัวเป็นเครือข่ายกัน และเชื่อมกันในลักษณะที่เรียกว่า ไฮเปอร์คิว

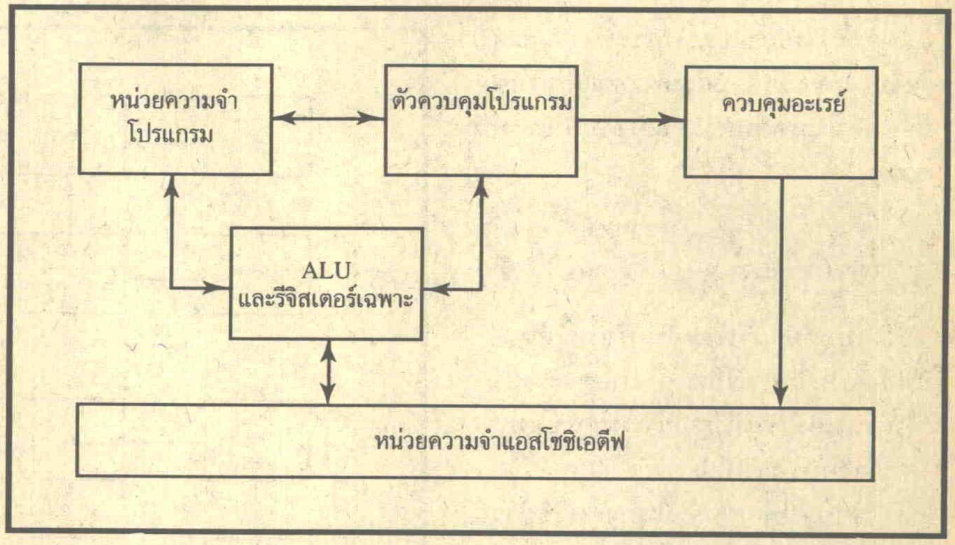
สถาปัตยกรรมแบบแอสโซซิเอตีฟเมมโมรีโปรเซสเซอร์ (Associative memory processor)

การต่อหน่วยความจำแบบนี้ทำให้สถาปัตยกรรมแตกต่างจากแบบ SIMD ไปบ้าง แนวความคิดในลักษณะนี้มีมาตั้งแต่ยุคเริ่มต้นของการสร้างคอมพิวเตอร์ ในแนวความคิดนี้เน้นวิธีการเรียกค้นหน่วยความจำแบบขนานเพื่อให้ข้อมูลเข้ามาประมวลผลได้หลายตัว หลังจากนั้นได้มีการพัฒนากันในรูปแบบสถาปัตยกรรมต่าง ๆ เพื่อให้การเรียกเข้า

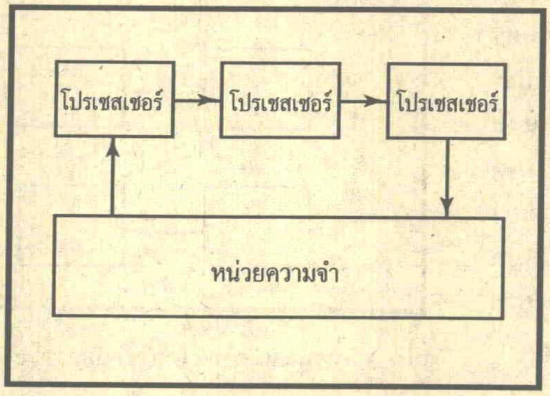
หาหน่วยความจำมีลักษณะเฉพาะ ซึ่งเน้นในเรื่องการใช้งานทางด้านฐานข้อมูลเป็นหลัก

รูปที่ 6 แสดงถึงฟังก์ชันการใช้แอสโซซิเอตีฟเมมโมรีกับโปรเซสเซอร์ตัวควบคุมโปรแกรม (program controller) เปรียบเสมือนเป็นคอมพิวเตอร์ที่ทำงานอนุกรมเพื่ออ่านและเอ็กซีคิวต์คำสั่ง แล้วส่งไปยังตัวควบคุมอะเรย์ และขณะเดียวกันก็จะไปเลือกหน่วยความจำที่ตรงกับการคำนวณ เพื่อนำข้อมูลมาประมวลผลใน ALU หรือในรีจิสเตอร์ รีจิสเตอร์เฉพาะบางตัวก็ไปอีนาเบิ้ลตัวควบคุมโปรแกรมและส่งต่อไปยังตัวควบคุมอะเรย์เพื่อเลือกหาหน่วยความจำที่พอเหมาะ

คอมพิวเตอร์ที่ใช้เทคโนโลยีนี้ในปัจจุบันได้แก่ การใช้เทคโนโลยีบิตสไลด์ ซึ่งวางตัวในแนวคอลัมน์ เป็นบิตที่ทำงานร่วมกับหน่วยความจำแบบแอสโซซิเอตีฟ เช่น มีหน่วยความจำขนาด 32768 บิต แต่เมื่อมองเป็นเวิร์ดได้ 4096 เวิร์ด ส่วนโปรเซสเซอร์ที่ทำงานในส่วนย่อยก็มีด้วยกัน 4096 ตัว



รูปที่ 6 โครงสร้างของแอสโซซิเอตีฟเมมโมรีโปรเซสเซอร์



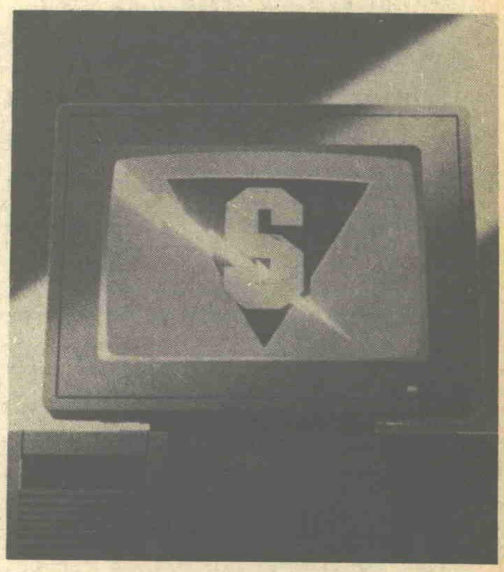
รูปที่ 7
การไหลของข้อมูลเป็นจังหวะ

สถาปัตยกรรมแบบซีสโตลิก (Systolic)

ในตอนต้นศตวรรษที่ 1980 H.T Kung แห่งมหาวิทยาลัยคาร์เนกี้เมลลอน ได้เสนอสถาปัตยกรรมแบบซีสโตลิก เพื่อใช้ในการแก้ปัญหาเฉพาะบางอย่างที่ต้องมีการคำนวณแบบแบ่งกับคำนวณอย่างสมดุลย์ โดยการเรียกใช้แถบกว้างของอินพุตเอาต์พุต ซีสโตลิกที่เสนอในขณะนั้นเป็นแบบไปป์ไลน์มัลติโปรเซสเซอร์ โดยข้อมูลที่เรียกเข้ามาประมวลผลจากหน่วยความจำจะได้รับการเรียกเข้ามาดำเนินการเป็นจังหวะผ่านทางเครือข่ายของโปรเซสเซอร์ ก่อนที่จะกลับไปยังหน่วยความจำอีกครั้ง รูปที่ 7 เป็นตัวอย่างของการเรียกข้อมูลผ่านโปรเซสเซอร์

การซิงโครไนซ์ข้อมูลระหว่างซีพียูกระทำโดยสัญญาณนาฬิกาจากภายนอก โปรเซสเซอร์แต่ละตัวสามารถรับส่งข้อมูลระหว่างกัน และรับส่งข้อมูลกับหน่วยความจำ การส่งผ่านข้อมูลให้กันนี้จึงเป็นวิธีการที่ทำให้เกิดการคำนวณแล้วส่งผลลัพธ์ระหว่างกัน

การทำงานแบบขนานที่ใช้หลักการสร้างจังหวะที่เรียกว่าซีสโตลิกนี้ จึงเป็นเสมือนการขนานกันเป็นกลุ่ม ๆ ที่อาศัยโปรเซสเซอร์หลายตัวทำงานร่วมกัน



เพื่อให้เข้าใจภาพการทำงานง่ายยิ่งขึ้น จะขอยกตัวอย่างการคูณแมตริกซ์ ซึ่งปกติมีขั้นตอนการคูณหลายขั้นตอน เช่น

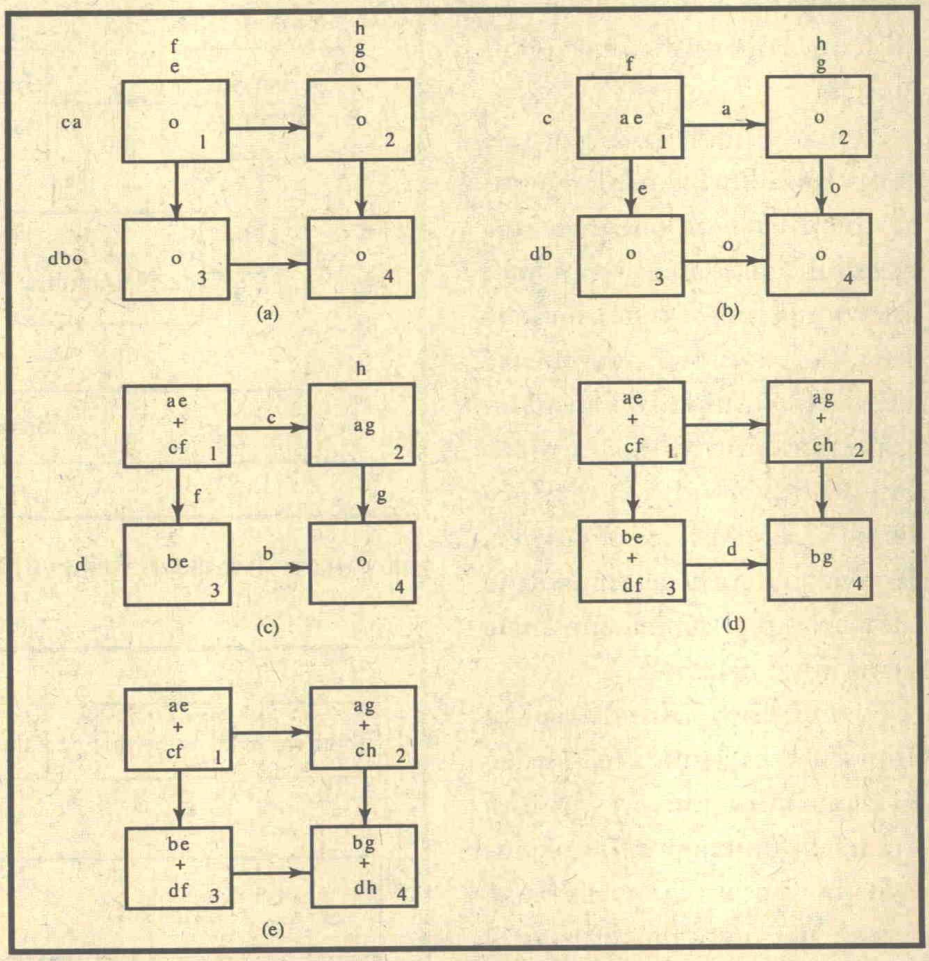
$$A = \begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad \text{และ} \quad B = \begin{vmatrix} e & f \\ g & h \end{vmatrix}$$

เมื่อต้องการคูณกันเพื่อให้ได้ผลลัพธ์ A.B ในกรณีนี้เราวางโปรเซสเซอร์เป็นแบบอะเรย์ที่มีการส่งผ่านข้อมูลระหว่างกันดังรูปที่ 8

จากรูปที่ 8 เป็นกรรมวิธีการทำงานที่ชัดเจน โดยเริ่มต้นมีโปรเซสเซอร์สี่ตัว โปรเซสเซอร์แต่ละตัวมีอินพุตเอาต์พุตต่อเชื่อมกัน และเชื่อมโยงกับข้อมูลเริ่มต้น ดังรูป a ทุกโปรเซสเซอร์มีผลลัพธ์ของการคำนวณเป็น 0 โปรเซสเซอร์ตัวที่ 1 รับอินพุต e f และ a c โปรเซสเซอร์ตัวที่ 2 รับอินพุต o g h และข้อมูลที่ส่งจากโปรเซสเซอร์ตัวที่ 1 โปรเซสเซอร์ตัวที่ 3 รับข้อมูลจากโปรเซสเซอร์ตัวที่ 1 และ o b d โปรเซสเซอร์ตัวที่ 4 รับข้อมูลจากโปรเซสเซอร์ตัวที่ 2 และ 3

เมื่อผ่านจังหวะที่ 1 โปรเซสเซอร์ทั้ง 4 ตัวทำงาน คำสั่งคูณผลลัพธ์จะได้ดังรูป b ขณะเดียวกันแต่ละโปรเซสเซอร์ก็จะส่งข้อมูลให้กัน เมื่อผ่านจังหวะที่ 2 ก็จะได้ดังรูป c การคูณจะเป็นการคูณแล้วบวกแบบสะสม ดังนั้นในโปรเซสเซอร์ตัวที่ 1 จึงมีข้อมูล ae+cf

ดังนั้นเมื่อผ่านไป 5 จังหวะก็จะทำให้การคูณแมตริกซ์เสร็จสมบูรณ์ ผลลัพธ์อยู่ในโปรเซสเซอร์แต่ละตัว การคูณแมตริกซ์ในลักษณะนี้จึงเป็นการเพิ่มความเร็วขึ้นจากเดิมได้อีกมาก สถาปัตยกรรมแบบซีสโตลิกจึงเป็นสถาปัตยกรรมที่มีประสิทธิภาพตามโครงสร้างการคำนวณแบบหนึ่ง เหมาะกับโครงสร้างการคำนวณที่ชัดเจนมีรูปแบบที่แน่นอน เช่น ตัวอย่างดังกล่าว การวางอะเรย์ของโปรเซสเซอร์และแต่ละตัวของโปรเซสเซอร์เชื่อมโยงกับข้อมูลในหน่วยความจำและส่งข้อมูลระหว่างโปรเซสเซอร์ด้วย ทำให้โมเดลการคำนวณทำได้รวดเร็วมาก



รูปที่ 8 การคูณแมตริกซ์โดยใช้หลักการซีสโตลิก

สถาปัตยกรรมแบบ MIMD

สถาปัตยกรรมแบบ MIMD เป็นสถาปัตยกรรมที่ใช้โปรเซสเซอร์จำนวนหลาย ๆ ตัว แต่ละตัวทำงานอิสระในแต่ละคำสั่งที่แยกกัน โดยใช้ข้อมูลที่แตกต่างกัน การทำงานจึงเป็นเสมือนการคำนวณทั่วไปที่จัดรูปแบบให้มาทำงานขนานกันในบางส่วน มีการส่งผ่านข้อมูลระหว่างโปรเซสเซอร์ เมื่อคำนวณเสร็จเรียบร้อยแล้วเพื่อซิงโครไนซ์ให้ได้คำตอบ

MIMD จึงเป็นเครื่องจักรที่สนับสนุนแนวทางการทำงานแบบขนานในระดับสูง ที่มีการแบ่งแยกงานเป็นโมดูลหรือการเขียนโปรแกรมเป็นส่วนให้แต่ละส่วนทางงานอิสระ แล้วนำผลลัพธ์มาร่วมกัน โดยการส่งข้อความข่าวสารถึงกัน

MIMD เป็นวิธีการคำนวณที่แตกงานใหญ่ให้เป็นงานย่อย แบ่งแยกและ

จัดลำดับงานให้ชัดเจนเหมือนการก่อสร้าง มีงานบางอย่างทำขนานกันได้เพื่อให้งานเสร็จสิ้นเร็วขึ้น

โครงสร้างของ MIMD ส่วนใหญ่จึงให้หน่วยความจำกับโปรเซสเซอร์เป็นชุด ๆ โปรเซสเซอร์แต่ละตัวจะมีหน่วยความจำของตนเอง เพื่อให้การทำงานในแต่ละโมดูลอิสระ

การแบ่งประเภทของสถาปัตยกรรม MIMD แบ่งได้หลายรูปแบบดังนี้

สถาปัตยกรรมแบบกระจายหน่วยความจำ

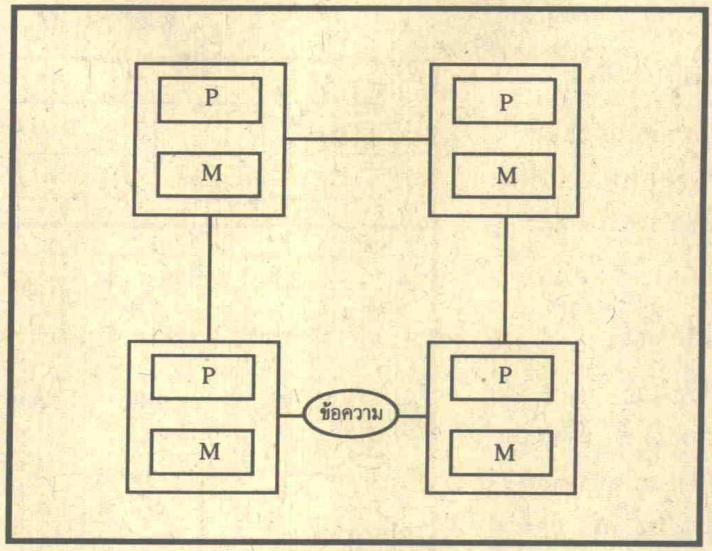
สถาปัตยกรรมแบบนี้มีใช้กันมาก แสดงโครงสร้างได้ดังรูปที่ 9 หน่วยความจำที่แยกได้แต่ละโปรเซสเซอร์เรียกว่า หน่วยความจำโลคอล ซึ่งจะใช้ได้เฉพาะซีพียูนั้น ๆ แต่มีการเชื่อมโยงระหว่างโปรเซสเซอร์เพื่อส่งข้อความหรือผลลัพธ์ระหว่างกัน การเชื่อมโยงระหว่างโปรเซสเซอร์จัดเป็นรูปแบบทางโทโปโลยีได้หลายรูปแบบ รูปที่ 10 แสดงรูปแบบที่ใช้ในระบบสถาปัตยกรรมแบบกระจายหน่วยความจำที่มีผู้นำเสนอและสร้างกันขึ้นมา

สถาปัตยกรรมรูปวงแหวน

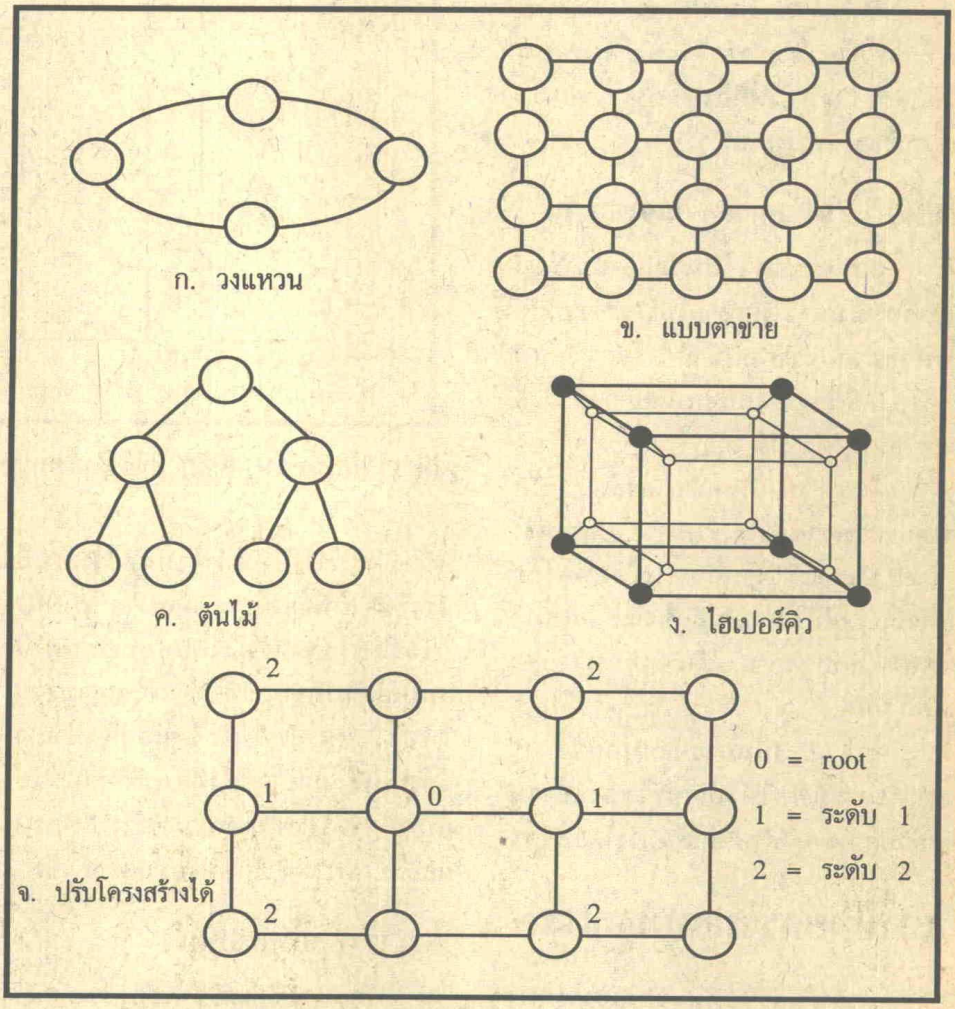
ในกรณีนี้ใช้หลักการของระบบแลนเพื่อส่งข้อมูลระหว่างโปรเซสเซอร์ในรูปแบบแยกเก็บที่มีขนาดจำกัดไประหว่างโปรเซสเซอร์เพื่อการทำงานซิงโครไนซ์กัน การใช้สถาปัตยกรรมแบบนี้ส่วนใหญ่จะใช้กับโปรเซสเซอร์ไม่มากนัก

สถาปัตยกรรมแบบตาข่าย

มีการจัดวางเป็นอะเรย์ ลักษณะเช่นนี้จะมีจำนวนโหนดเป็น n^2 โหนด แต่ละโหนดหรือโปรเซสเซอร์แต่ละตัวจะเชื่อมต่อกับตัวข้างเคียง



รูปที่ 9. โครงสร้างของ MIMD มีการแยกหน่วยความจำออกไปยังโปรเซสเซอร์



รูปที่ 10 การเชื่อมโยงหลากหลายโปรเซสเซอร์ของ MIMD

สถาปัตยกรรมแบบต้นไม้

เป็นการส่งผ่านข้อมูลแบบกระจายตามลำดับขั้น การส่งข้อมูลหว่างกันกระจายไปกลับตามเส้นทางของแต่ละโหนดที่เชื่อมต่อกัน

สถาปัตยกรรมแบบไฮเปอร์คิว

ลักษณะนี้บางทีเรียกว่า n คิว คือวางเป็นลูกบาศก์ซ้อนกัน การเชื่อมโยงแบบนี้ทำให้แต่ละโปรเซสเซอร์เชื่อมกับซีพียูข้างเคียงเป็นจำนวน n ลิงค์ เช่น ในรูปมีจำนวน 4 ลิงค์เป็นต้น

สถาปัตยกรรมที่ปรับเปลี่ยนรูปได้

เป็นการวางแบบอะเรย์เพื่อให้มีการเชื่อมโยงตามการคำนวณที่กำหนดในแต่ละปัญหา เพื่อกำหนดรูปแบบการเชื่อมโยงที่เหมาะสม

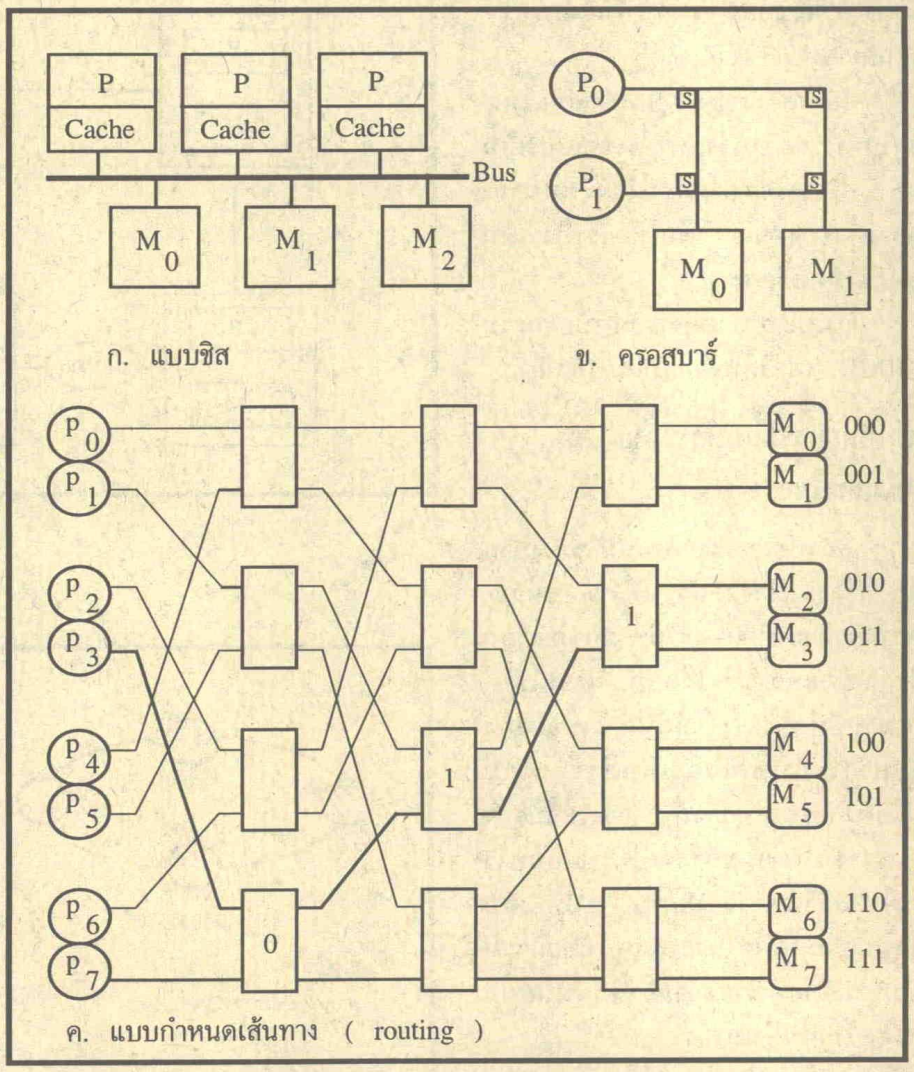
MIMD แบบใช้หน่วยความจำร่วมกัน

ลักษณะการใช้หน่วยความจำร่วมกันคือ เป็นการให้หลายโปรเซสเซอร์ใช้หน่วยความจำร่วมกัน มีการจัดรูปแบบโทโปโลยีของการแบ่งหน่วยความจำได้หลายแบบดังรูปที่ 11

ในรูป ก. เป็นแบบโครงสร้างบัสที่ทุกโปรเซสเซอร์วิ่งเข้าหาหน่วยความจำ

รูป ข. ใช้หลักการเชื่อมต่อโดยการมีสวิตช์ปิดเปิดเป็นลักษณะครอสบาร์ เพื่อให้โปรเซสเซอร์กับหน่วยความจำต่อถึงกัน

รูป ค. เป็นการกำหนดด้วยการหาเส้นทางเพื่อให้โปรเซสเซอร์ใช้งานหน่วยความจำ กำหนดเส้นทางโดยสวิตช์



รูปที่ 11 โครงสร้าง MIMD ที่มีการแบ่งหน่วยความจำใช้ร่วมกัน

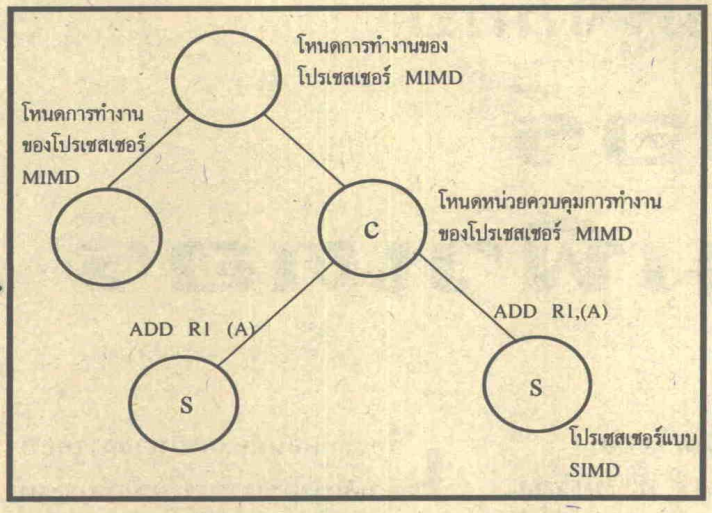
สถาปัตยกรรมแบบลูกผสม

การนำเอาข้อดีทั้งของ MIMD และ SIMD มาผสมกันเป็นหนทางหนึ่งที่จะทำให้การทำงานของระบบมีประสิทธิภาพดียิ่งขึ้น ทั้งนี้เพราะ MIMD มีโครงสร้างการทำงานแยกออกเป็นแบบอะซิงโครนัสได้ กล่าวคือ สามารถให้ทำงานบางส่วนเสร็จแล้วจึงนำเอาข้อมูลหรือผลลัพธ์ส่งต่อมาทำงานร่วมกัน การจัดรูปแบบผสมระหว่าง MIMD และ SIMD ทำได้หลายรูปแบบ เช่น

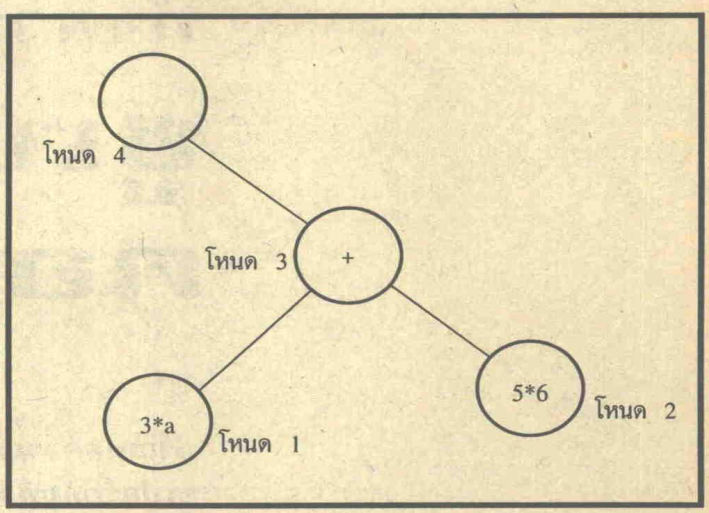
โครงสร้าง MIMD/SIMD

เป็นการสร้างเครือข่ายโปรเซสเซอร์ที่มีบางส่วนเป็นแบบ MIMD และบางส่วนเป็นแบบ SIMD ลักษณะการทำงานจึงมีส่วนการควบคุมที่เราเรียกว่า ตัวควบคุมโหนด เพื่อให้ภาพการทำงานที่เข้าใจได้ชัดเจนขึ้น ขอแสดงการทำงานเป็นไดอะแกรมดังรูปที่ 12 ลักษณะนี้จะมีส่วนที่ทำงานแบบ SIMD ทำงานพร้อมกันภายใต้การควบคุมของโหนดใดโหนดหนึ่ง

สังเกตว่าลักษณะกลุ่มการทำงานจะขึ้นกับการจัดโครงสร้างระบบงานเพื่อแยกแยะส่วนการทำงานในระดับต่าง ๆ ให้ทำงานตามลักษณะพิเศษ เช่น



รูปที่ 12 การทำงานแบบ MIMD/SIMD



รูปที่ 13 การกระจายแบบดาต้าโฟลว์

กลุ่ม SIMD ทำงานเพื่อให้ได้ผลลัพธ์ที่ต้องการ ส่วนการทำงาน MIMD ก็แยกการทำงาน คอมพิวเตอร์แบบนี้ต้องมีการแบ่งงานและการกระจายงานที่ดีและเหมาะสม

สถาปัตยกรรมดาต้าโฟลว์

สถาปัตยกรรมแบบนี้เป็นที่รู้จักและมีการประยุกต์มาใช้กันนานแล้ว เป็นวิธีการกระจายการทำงานเพื่อให้ส่วนการทำงานที่ขนานกันได้ เพื่อลดเวลาการทำงานโดยรวมลง สถาปัตยกรรมแบบนี้อธิบายได้โดยการแก้ปัญหาง่าย ๆ เช่น 3*a + 5*b ซึ่งเห็นว่าข้อมูลในส่วน 3*a และ 5*b ทำงานพร้อมกันหรือขนานกันได้ เมื่อได้ผลลัพธ์จึงนำมาบวกกันต่อไป

การเขียนแบบกระจายข้อมูลให้ทำงานขนานกันแบบดาต้าโฟลว์เขียนเป็นรูปภาพได้ดังรูปที่ 13

การกระจายข้อมูลและทำการคำนวณแยกกัน เพื่อนำผลลัพธ์มาดำเนินการเป็นวิธีการใช้หลักการที่สร้างโปรเซสเซอร์แบบผสมได้

สถาปัตยกรรมแบบขนานจึงเป็นรูปแบบสำหรับสร้างคอมพิวเตอร์ประสิทธิภาพสูง

จากที่กล่าวมาทั้งหมดเป็นเพียงหลักการของเทคโนโลยีที่จะทำให้คอมพิวเตอร์ที่พัฒนาขึ้นพ้นภาวะข้อจำกัดทางด้านการทำงานที่อยู่ภายใต้ตัวชิพ เช่น สัญญาณนาฬิกา หรือความเร็วในเชิงคำนวณ การสร้างโครงสร้างแบบขนานทำให้การทำงานขนาดใหญ่ที่ต้องการความซับซ้อนสูงมาก สามารถทำได้ในเวลาที่สั้นกว่า

ประจวบกับเทคโนโลยีการสร้างโปรเซสเซอร์และหน่วยความจำทำได้ดีและสามารถสร้างซีพียูจำนวนมากได้ในราคาถูก ดังนั้นการสร้างรูปแบบที่คอมพิวเตอร์มีซีพียูแบบขนานจึงกำลังได้รับความสนใจและพัฒนาอยู่อย่างเข้มแข็งและจริงจัง

เอกสารอ้างอิง

1. Ralph Duncan "A Survey of Parallel Computer Architecture" IEEE Computer Magazine Feb 1990.